

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی برق

نام دانشجو: کسری خلفی

شماره ی دانشجویی : ۹۵۲۳۰۳۸

استاد درس : دکتر فرزانه عبداللهی

سرپرست آزمایشگاه: مهندس امینی

آزمایش شماره ی چهارم

پایتون:

هدف در این آزمایش پیاده سازی روش K-Mean بود

در این آزمایش چندین تابع تعریف کردیم که به صورت زیر میباشد

- تابع فاصله: این تابع با دریافت دو عدد فاصله ی اقلیدسی این دو عدد را حساب میکند

```
6 def distance(center, input):
7     sum = 0
8     for i in range(len(input)):
9         sum += (center[i]-input[i])**2
10    return np.sqrt(sum)
11
```

- تابع مرکز: با گرفتن تعدادی ورودی مرکز این داده ها را حساب میکند

```
12
13 def make_center(input: np.ndarray):
14     sum = np.zeros((1, len(input[0])))
15     for i in range(len(input)):
16         sum += input[i]
17     return sum/len(input)
18
```

- تابع ماسک: این تابع با گرفتن داده ها و همچنین مراکز مشخص شده در تابع بالایی مشخص میکند که هر داده به کدام کلاس متعلق میباشد.

```
20 def make_mask(data, centers):
21     mask = np.zeros((len(data), 1))
22     for i in range(len(data)):
23         min_dist = 10000000
24         index = -1
25         for j in range(len(centers)):
26             if distance(centers[j], data[i]) < min_dist:
27                 min_dist = distance(centers[j], data[i])
28                 index = j
29         mask[i] = index
30     return mask
31
```

کد با توجه به توابع تعریف شده در بالا به صورت زیر می باشد:

```
while True:

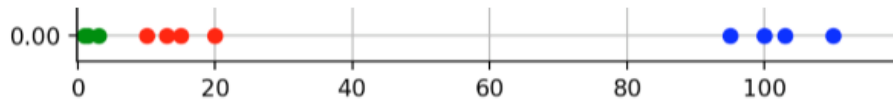
    mask = make_mask(data, centers)

    for i in range(k):
        my_list = []
        for j in range(len(mask)):
            if mask[j] == i:
                my_list.append(data[j])
        #print(np.array(my_list))
        if len(my_list) == 0:
            return k_Mean(data, k)
        centers[i] = make_center(np.array(my_list))
    error = 0
    for i in range(len(centers)):
        error += distance(centers[i], last_centers[i])
    if error < 1:
        return mask, centers
    last_centers = centers
```

کد بالا برای تمام ورودی ها کار میکند حال به ترتیب ورودی های یک و دو بعدی به تابع میدهیم:

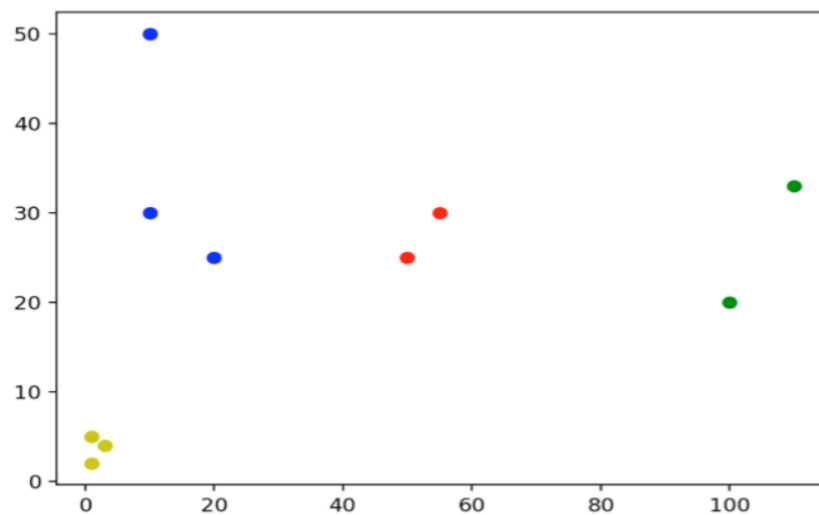
آزمایش اول - یک بعدی

```
data = np.array([[1], [3], [1.5], [10], [13], [100], [110], [103], [95], [20], [15]])
k = 3
mask, centers = k_Mean(data, k)
fig, ax = plt.subplots()
template = ['ro', 'bo', 'go', 'yo', 'r*', 'b*', 'g*', 'y*']
for i in range(len(data)):
    ax.plot(data[i][0], 0, template[int(mask[i])])
plt.grid()
plt.show()
```



آزمایش دوم - دو بعدی

```
data = np.array([[1, 2], [3, 4], [1, 5], [10, 30], [10, 50], [100, 20], [20, 25], [10, 50], [55, 30], [50, 25], [110, 33]])
k = 4
mask, centers = k_Mean(data, k)
fig, ax = plt.subplots()
template = ['ro', 'bo', 'go', 'yo', 'r*', 'b*', 'g*', 'y*']
for i in range(len(data)):
    ax.plot(data[i][0], data[i][1], template[int(mask[i])])
plt.show()
```



در ادامه برای تست این روش عکس معروف پردازش تصویر را به عنوان ورودی به کد بالا دادیم و تصویر خروجی سیاه سفید به ما داد

```
lena = cv2.imread('lena.jpg', 0)
lena = np.reshape(lena, (1, -1))
lena = np.transpose(lena)

k = 5
mask, centers = k_Mean(lena,k)
for i in range(len(lena)):
    | lena[i] = centers[int(mask[i])]

lena = np.transpose(lena)
lena = np.reshape(lena, (446, 651))
cv2.imwrite("mylena.png", lena)
cv2.imshow(lena)
```



تمرین

۱. با توجه به نمودار خطا بر حسب تعداد کلاس ها که در قسمت ارزیابی الگوریتم رسم کرده اید حالت بهینه الگوریتم در چه تعداد کلاس رخ می دهد؟

جواب قطعی برای تعیین تعداد بهینه خوشه ها در الگوریتم های خوشه بندی وجود ندارد! تعیین تعداد بهینه خوشه ها از سویی تابع معیار در نظر گرفته شده به عنوان معیار شباهت بین داده ها در الگوریتم خوشه بندی است و از سوی دیگر تابع پارامترهای تنظیم شده در الگوریتم موردنظر است. یک روش ساده و رایج بین کاربران، بررسی دندروگرام حاصل از یک الگوریتم خوشه بندی سلسله مراتبی مانند Agglomerative است تا از این طریق، در صورت حصول یک مقدار خاص برای خوشه ها از آن تعداد به عنوان مقدار k در الگوریتم های خوشه بندی مبتنی بر تقسیم استفاده شود. اما این روش، فاقد پشتوانه ریاضی و اثباتی قوی است.

۲. به نظر شما چرا الگوریتم توانایی دسته بندی صحیح داده ها را در قسمت محدودیت های $k - mean$ نداشت؟ برای بهبود الگوریتم چه پیشنهادی دارید؟

در الگوریتم k mean در صورتیکه مقادیر اولیه مرکز کلاس ها به خوبی انتخاب نشوند ممکن است هیچ data یی به آن محول نشود و درواقع یکی از کلاس ها از بین برود.